

Approximation of Event Probabilities in Noisy Cellular Processes

Frédéric Didier¹, Thomas A. Henzinger¹, Maria Mateescu¹, and Verena Wolf^{1,2}

¹EPFL, Switzerland

²Saarland University, Germany

Abstract. Molecular noise, which arises from the randomness of the discrete events in the cell, significantly influences fundamental biological processes. Discrete-state continuous-time stochastic models (CTMC) can be used to describe such effects, but the calculation of the probabilities of certain events is computationally expensive.

We present a comparison of two analysis approaches for CTMC. On one hand, we estimate the probabilities of interest using repeated Gillespie simulation and determine the statistical accuracy that we obtain. On the other hand, we apply a numerical reachability analysis that approximates the probability distributions of the system at several time instances. We use examples of cellular processes to demonstrate the superiority of the reachability analysis if accurate results are required.

1 Introduction

The traditional approach for a dynamical model of cellular reaction networks is based on the assumption that the concentrations of the chemical species change continuously and deterministically in time. During the last decade, however, stochastic models with discrete state spaces have seen growing interest [25, 30, 36, 7, 29, 38, 40, 43]. The reason is that they take into account the effects of molecular noise in the cell. Molecular noise has a significant influence on important processes such as gene expression [19, 3, 27, 24, 6, 39], decisions of the cell fate [1, 23, 22], and circadian oscillations [11, 2, 12].

An appropriate modeling approach for systems that are subject to molecular noise is a discrete-state continuous-time Markov process, also called *continuous-time Markov chain* (CTMC). This is particularly evident in the presence of *intrinsic noise* arising from random microscopic events in the cell, such as the location of molecules or the order of the reactions. As opposed to continuous models, the discrete-state stochastic model is able to capture the discreteness of the random events in the cell.

The evolution of such a CTMC is given by a master equation that is derived according to Gillespie’s theory of stochastic chemical kinetics [10]. Since the state space grows exponentially in the number of involved chemical species, the state space of the CTMC is large, which renders its analysis difficult. Moreover, the discrete structure becomes even larger when the number of molecules in the system grows. If the populations of certain chemical species are large, their effect on the system’s variance is small and they can be approximated assuming a continuous deterministic change. For species with small populations, however, a continuous approximation is not appropriate

and other approximation techniques are necessary to reduce the computational effort of the analysis.

Besides the computation of cumulative measures such as expectations and variances of the populations of certain chemical species, the computation of event probabilities is important for several reasons. First, cellular process may decide probabilistically between several possibilities, e.g., in the case of developmental switches [14, 1, 30]. In order to verify, falsify, or refine the mathematical model based on experimental data, the likelihood for each of these possibilities has to be calculated. But also full distributions are of interest, such as the distribution of switching delays [24], the distribution of the time of DNA replication initiation at different origins [28], and the distribution of gene expression products [42]. Finally, many parameter estimation methods require the computation of the posterior distribution because means and variances do not provide enough information to calibrate parameters [16].

Two different families of computational approaches have been proposed and used to estimate event probabilities and approximate probability distributions. The first kind of approach is based on numerical simulation, i.e., the generation of many sample trajectories (or *simulation runs*) of the system. The second kind of approach is based on numerical reachability analysis, i.e., the propagation of the probability mass through the state space. The former approach is known as *Gillespie simulation* [9], in which pseudo-random numbers are used to simulate molecular noise. Measures of interest are obtained via statistical output analysis. The main advantage of simulation is that it is easy to implement and the generation of trajectories is not limited by the size of the state space. Moreover, the precision level of the method can be easily adjusted by performing more or fewer simulation runs. For the computation of the probability of certain events, however, simulative approaches become computationally expensive, because a large number of runs have to be carried out to bound the statistical error appropriately. For estimating event probabilities, a higher precision level is necessary than for estimating cumulative measures such as expectations, and simulation becomes expensive because doubling the precision requires four times more simulation runs.

In contrast, approaches based on a numerical reachability analysis approximate probability distributions of the CTMC. As opposed to a statistical estimation of probabilities, which yields an indirect solution, the master equation is numerically solved by integrating the system's behavior over time. Standard numerical techniques are impractical for many systems because of the enormous size of the state space. Recently, however, more sophisticated numerical approximation methods have been proposed, which solve the system in an iterative fashion and consider only subsets of the state space during any given time interval [17, 26, 5]. They are significantly more efficient than global analysis because they use localization optimizations (such as "sliding windows") and dynamic adaptation ("on-the-fly" generation of windows). These methods efficiently compute the probability distribution of large CTMC at several time instances up to a small approximation error. They can also be used for infinite-state systems.

In this paper, we evaluate and compare the performance of the two different approaches for the computation of probabilities of certain events, i.e., the statistical estimation using simulation and the approximation using a numerical reachability analysis. For the latter we use a particular algorithm as a representative of the whole family of

numerical analysis algorithms, because we have found it to perform best. Similar to the sliding-window method [17], our algorithm performs a sequence of local analysis steps on dynamically constructed abstractions of the system. The main improvement over the sliding-window method is that our algorithm is based on adaptive uniformization [41], which allows us to consider arbitrary sets of significant states, i.e., they may be located at different parts of the state space and are not restricted to a specific window shape. Moreover, adaptive uniformization is more robust if the system under study is stiff, i.e., if the chemical reactions occur at time scales that differ by several orders of magnitude. In contrast to [17], here, for the first time, we perform a systematic experimental performance comparison of a numerical reachability analysis with simulation.

The first example that we consider is the transcription regulation of a repressor protein in bacteriophage λ [13], where we approximate the probability distribution at several time instances. In the second example, which is a gene expression network [39], we compute the distribution of the time until the number of produced proteins exceeds a certain threshold. In both examples the number of states reachable from the initial state is infinite. The number of chemical species is 6 and 2, and the number of chemical reactions is 10 and 4, respectively. We compare the running time of our numerical reachability analysis to that of the simulative approach for both examples, for different precision levels. Our results show that numerical approximation based on reachability analysis is superior to statistical estimation based on repeated simulation, especially if we increase the desired precision level. For instance, the numerical approximation of the first example needs 39 minutes for a total approximation error of 2×10^{-5} , which distributes among all states. Simulation requires more than six hours if the statistical error of a single event is to be bounded by 10^{-5} and more than sixty hours for 10^{-6} .

2 Stochastic Model

According to the theory of stochastic chemical reaction kinetics, a continuous-time Markov chain (CTMC) can be derived from a set of biochemical reactions [18, 10]. This discrete-state model has a regular structure, which gives rise to a functional description in terms of *transition class models* (TCMs) [33]. TCMs naturally represent coupled chemical reactions as each chemical reaction corresponds to a transition class. They provide, however, a more general description than a set of chemical reactions.

2.1 Transition Class Models

Consider a dynamical system with a finite number of discrete state variables such as the number of instances of some chemical species in a reaction volume. Assume that these variables change at discrete points in time. A *transition class* provides a rule for these changes and a function for the calculation of the state-dependent *transition rate* at which a state change occurs. Let S be a countable set of states.

Definition 1. A transition class C is a triple (G, u, α) such that (i) the guard $G \subset S$ is a subset of S , (ii) $u : G \rightarrow S$ is an injective update function with $u(x) \neq x$ for all $x \in G$, (iii) $\alpha : G \rightarrow \mathbb{R}_{>0}$ is a rate function. A transition class model (TCM) $M = (y, \{C_1, \dots, C_k\})$ consists of an initial state $y \in S$ and a finite set of transition classes C_1, \dots, C_k .

The set G contains all states x in which a transition of type C is possible and $u(x)$ is the target state of the transition. The probability of the C -transition depends on the transition rate $\alpha(x)$ in the way explained below.

In practice, we can usually express G by a finite number of constraints on the state variables, and u and α by elementary arithmetic functions. Thus, a TCM provides a finite description of a (possibly infinite-state) system. Before we show how a CTMC is derived from a TCM, we present some examples of TCMs that describe biochemical reaction networks.

Biochemical Reaction Networks. We consider a fixed reaction volume with n different chemical species that is spatially homogeneous and in thermal equilibrium. Then, the state space of the system is given by $S = \mathbb{N}_0^n$. We assume that molecules collide randomly and that collisions may lead to chemical reactions. For a given set of chemical reactions, we construct a TCM such that each transition class corresponds to a reaction and the associated propensity function is given by the rate function α .

Example 1. We consider a simple transition class model for transcription of a gene into messenger RNA (mRNA), and subsequent translation of the latter into proteins [39]. This reaction network involves three chemical species, namely, gene, mRNA, and protein. As only a single copy of the gene exists, a state of the system is uniquely determined by the number of mRNA and protein molecules. Therefore, $S = \mathbb{N}_0^2$ and a state is a pair $(x_R, x_P) \in S$. We assume that initially there are no mRNA molecules and no proteins in the system, i.e., $y = (0, 0)$. The following four types of reactions occur in the system, namely $\emptyset \rightarrow mRNA$, $mRNA \rightarrow mRNA + P$, $mRNA \rightarrow \emptyset$, and $P \rightarrow \emptyset$. Let $i \in \{1, \dots, 4\}$ and let $c_i > 0$ be a constant. Transition class $C_i = (G_i, u_i, \alpha_i)$ describes the i -th reaction type.

- We describe gene transcription by transition class C_1 , which increases the number of mRNA molecules by 1. Thus, $u_1(x_R, x_P) = (x_R + 1, x_P)$. This transition class is possible in all states, i.e., $G_1 = S$. Transcription happens at the constant rate $\alpha_1(x_R, x_P) = c_1$, as only one reactant molecule (the gene) is available.
- We represent the translation of mRNA into protein by C_2 . A C_2 -transition is only possible if there is at least one mRNA molecule in the system. We set $G_2 = \{(x_R, x_P) \in S \mid x_R > 0\}$ and $u_2(x_R, x_P) = (x_R, x_P + 1)$. Note that in this case mRNA is a reactant that is not consumed. The translation rate depends linearly on the number of mRNA molecules. Therefore, $\alpha_2(x_R, x_P) = c_2 \cdot x_R$.
- Degradation is modeled by C_3 and C_4 . Hence, $G_3 = G_2$, $G_4 = \{(x_R, x_P) \in S \mid x_P > 0\}$, $u_3(x_R, x_P) = (x_R - 1, x_P)$, and $u_4(x_R, x_P) = (x_R, x_P - 1)$. We set $\alpha_3(x_R, x_P) = c_3 \cdot x_R$ and $\alpha_4(x_R, x_P) = c_4 \cdot x_P$.

2.2 Chemical Master Equation

A transition class model $M = (y, \{C_1, \dots, C_k\})$ represents a time-homogeneous, discrete-state Markov process $\{X(t)\}_{t \geq 0}$, that is, a CTMC with state space S . The j -th entry of the random vector $X(t) = (X_1(t), \dots, X_n(t))$ represents the value of the j -th state variable. Let $C_m = (G_m, u_m, \alpha_m)$, $1 \leq m \leq k$, and assume that at time $t \geq 0$ the process is in state $x \in G_m$.

The probability of a transition of type C_m occurring in the next infinitesimal time interval $[t, t + \tau)$, $\tau > 0$ is given by

$$Pr(X(t + \tau) = u_m(x) \mid X(t) = x) = \alpha_m(x) \cdot \tau.$$

Since y is the initial state of M we have $Pr(X(0) = y) = 1$, and for $x \in S$ we define the probability that X is in state x at time t by $p^{(t)}(x) = Pr(X(t) = x \mid X(0) = y)$. Recall that u_m is injective. To simplify our presentation, we define the set H_m as the set of all states x for which $u_m^{-1}(x)$ is defined, that is, that can be reached by a transition of type C_m . The *chemical master equation* describes the behavior of X by the differential equation [18]

$$\frac{\partial p^{(t)}(x)}{\partial t} = \sum_{m: x \in H_m} \alpha_m(u_m^{-1}(x)) \cdot p^{(t)}(u_m^{-1}(x)) - \sum_{m: x \in G_m} \alpha_m(x) \cdot p^{(t)}(x). \quad (1)$$

Unbounded Range. For realistic systems, the state space of the Markov chain is extremely large, because its size grows exponentially in the number of involved chemical species. Moreover, if upper bounds on the state variables cannot be derived from certain conservation laws, their range is assumed to be infinite although in practice the number of molecules is bounded. Then from the infinite structure, we can compute bounds that are kept with a very high probability. Even though every state in the infinite state space has a non-zero probability, certain attracting regions force most of the probability mass to remain within a finite range.

Example 2. In Ex. 1, the degradation rates $\alpha_3(x)$ and $\alpha_4(x)$ grow linearly in the state variables. Thus, the higher the number of mRNA or protein molecules the more likely is their degradation. Depending on the rate constants c_1, \dots, c_4 , the system becomes “stable” in different regions. As time approaches infinity, the main part of the probability mass will be close to a region where production and degradation of molecules cancel each other out. Below, we discuss in general under which conditions the system approaches such a stable distribution.

Holding Times and Jump Probabilities. A Markov chain $\{X(t)\}_{t \geq 0}$ defined in the way above is a *stable and conservative jump process* [4]. Thus, there exists a sequence of jump times $\{\tau(n)\}_{n \geq 0}$ and a sequence $\{\hat{X}(n)\}_{n \geq 0}$ of visited states such that

$$\tau(0) = 0 < \tau(1) < \tau(2) < \dots \text{ and } X(t) = \hat{X}(n) \text{ if } \tau(n) \leq t < \tau(n+1).$$

The distribution of the n -th holding time $\tau(n+1) - \tau(n)$ under the condition $\hat{X}(n) = x$ is negative exponentially distributed with parameter $\lambda(x) = \sum_{m: x \in G_m} \alpha_m(x)$, also called *exit rate* of state x .

If the sum of all holding times is finite with positive probability, the Markov chain is said to *explode* and the limiting distribution does not exist. Explosive Markov chains are not of interest for the application area of this work since in this case the system “gets lost at infinity”. It is possible to check if the Markov chain does not explode by using *Reuter’s Criterion* [4]. For the remainder of our presentation we assume that the rate functions α_m are such that the Markov chain does not explode.

Assume that the n -th state of the Markov chain is x , that is, $\hat{X}(n) = x$. If at least one transition class is enabled in x , the successor state is $u_m(x)$ for some m with $x \in G_m$. The probability of successor $u_m(x)$ is given by

$$Pr(\hat{X}(n+1) = u_m(x) \mid \hat{X}(n) = x) = \frac{\alpha_m(x)}{\lambda(x)}.$$

The holding times and the jump probabilities play an important role for the simulation of the Markov chain, which is used to estimate the probability of a certain events.

3 Statistical Estimation of Probabilities

In this section we shortly review the basic steps that have to be carried out to estimate the probability of a certain measurable event using stochastic simulation. Throughout this section, we will denote this event by A and its probability by γ . For the analysis of biological systems, the events of interest may be the marginal distributions or even the joint distributions of certain chemical species. For instance, A may have the form $X_j(t) = k$, that is, the number of type j molecules is k .

Estimates are obtained in two steps. In the first step, a certain number of simulation runs of the Markov chain have to be generated, and in the second step, the results of the simulation runs are analyzed.

3.1 Trajectory Generation

A realization of the Markov chain, also called *trajectory* or *run*, is the random sequence of states visited by the process. If trajectories are produced by a computer, *pseudo-random numbers* are used to artificially generate randomness [20]. The basic steps of producing a single trajectory that starts in the initial state y at time 0 are as follows:

1. Initialize time $t = 0$ and state $x = y$.
2. Generate the holding time h , i.e., a sample of a random variable being exponentially distributed with parameter $-\lambda(x)$.
3. Generate the successor state, i.e., a sample m of a discrete random variable Z that has probability distribution $P(Z = m) = \alpha_m(x)/\lambda(x)$.
4. Set $t = t + h$, $x = u_m(x)$ and go to Step 2 if $t < T$.

In Step 2, we generate the holding time of the current state x . Pseudo-random number generators usually draw from a uniform distribution. Thus, for a given random sample r_1 that is uniformly distributed on $(0, 1)$, we calculate an exponentially distributed sample by using the inverse transform method. More precisely, we compute the inverse $-\frac{\ln r_1}{\lambda(x)}$ of the cumulative distribution function of the exponential distribution. In Step 3, the same idea is used to decide, which reaction occurs next. The inverse of the cumulative distribution function of Z is given by $m = \min\{i : \sum_{j=1}^i \alpha_j(x) > r_2 \cdot \lambda(x)\}$, where r_2 is again a random sample that is uniformly distributed on $(0, 1)$. In the final step, the current time and the current state are updated. The simulation is terminated if the time horizon T of interest is reached and continued otherwise.

3.2 Output Analysis

The problem of estimating the probability γ of the event A can be reformulated as estimating the expectation of the random variable χ_A with

$$\chi_A(\omega) = \begin{cases} 1 & \text{if } \omega \in A, \\ 0 & \text{if } \omega \notin A, \end{cases}$$

where ω is a trajectory. The expectation $E[\chi_A]$ equals γ , since $E[\chi_A] = 1 \cdot Pr(\chi_A = 1) + 0 \cdot Pr(\chi_A = 0) = \gamma$. Therefore, we can resort to the standard estimation procedure for expectations. Assume that N is the number of runs that have been carried out and Y_1, \dots, Y_N are independent and identically distributed as χ_A . Thus, from the i -th run we get a realization of Y_i by checking if A has occurred or not. It is important to point out that we have to guarantee the independence of the Y_i 's. This implies that we generate N independent trajectories of the Markov chain, each time with a different initial seed¹ for the pseudo-random number generator. The sample mean $\bar{Y} = \frac{1}{N} \sum_{i=1}^N Y_i$ is then an *unbiased* and *consistent estimator* [20] for $E[\chi_A]$. The former means that $E[\bar{Y}] = E[\chi_A]$ and the latter refers to the fact that as N increases the estimator \bar{Y} becomes closer to γ . Note that \bar{Y} is equal to the relative frequency of the event A . Let $\sigma^2 = VAR[\chi_A]$ be the variance of χ_A . We evaluate the quality of the estimator \bar{Y} by applying the central limit theorem, which states that \bar{Y} will approximately have a Normal distribution with mean $E[\chi_A] = \gamma$ and variance σ^2/N . Hence, for large N the random variable

$$Z = \frac{\bar{Y} - \gamma}{\sqrt{\sigma^2/N}}$$

has a standard Normal distribution, that is, the mean is zero and the variance is one. Knowing the distribution of Z enables us reason about the difference $|\bar{Y} - \gamma|$. Let $\beta \in [0, 1]$ be the *confidence level* and $z \in \mathbb{R}^+$ such that $\beta = Pr(|Z| \leq z)$. Then

$$\beta = Pr(|Z| \leq z) = Pr\left(\frac{|\bar{Y} - \gamma|}{\sqrt{\sigma^2/N}} \leq z\right) = Pr(|\bar{Y} - \gamma| \leq z\sqrt{\sigma^2/N}).$$

We estimate σ^2 with the sample covariance $S^2 = \frac{1}{N-1} \sum_{i=1}^N (Y_i - \bar{Y})^2$, which is an unbiased estimator for σ^2 . Then, for large N and a large number of realizations of the *confidence interval*

$$[\bar{Y} - z\sqrt{S^2/N}, \bar{Y} + z\sqrt{S^2/N}], \quad (2)$$

β is the fraction of intervals that cover γ . It therefore measures the quality of the estimator \bar{Y} .

For a practical application, two further remarks are important. Firstly, we usually choose $\beta \in \{0.95, 0.99\}$ and the corresponding value of z can be found in the table of the standard Normal distribution. Let Φ be the cumulative distribution function of the standard Normal distribution. Then, using that the Normal distribution is symmetric,

$$\Phi(z) = Pr(Z \leq z) = 1 - \frac{1-\beta}{2} = \frac{1+\beta}{2} \iff z = \Phi^{-1}\left(\frac{1+\beta}{2}\right).$$

Secondly, both, \bar{Y} and S^2 can be computed efficiently if during the trajectory generation the realizations of the two sums $\sum_{i=1}^N Y_i$ and $\sum_{i=1}^N Y_i^2$ are calculated, since it can be easily shown that

$$S^2 = \frac{\sum_{i=1}^N Y_i^2}{N-1} - \frac{(\sum_{i=1}^N Y_i)^2}{(N-1)N}.$$

¹ The seed of a pseudo-random number generator is an initial value, on which the sequence of generated numbers depend [20].

Thus, if $r \in \{0, \dots, N\}$ is the number of times event A occurred during the N simulation runs, $\bar{Y} = r/N$ and $S^2 = \frac{r(N-r)}{N(N-1)}$.

If the interval in Eq. 2 is large relative to \bar{Y} the quality of the estimator is poor and more simulation runs have to be carried out. For our experimental results in Section 5, we fixed the relative width of the interval to be 0.2 (which means that we have a relative error of at most 0.1) and chose confidence level $\beta = 0.95$. Thus, $z \approx 1.96$ and we can determine the number of necessary runs by bounding the relative width

$$2 \cdot \frac{z \cdot \sqrt{S^2/N}}{\gamma} \leq 0.2 \implies \frac{z^2}{0.01} \frac{S^2}{\gamma^2} \leq N \implies 384 \cdot \frac{S^2}{\gamma^2} \leq N$$

Assume now that we want to estimate the probability of events that occur at least with probability γ . Using the fact that $\sigma^2 = \text{VAR}[\chi_A] = \gamma(1-\gamma)$ and replacing S^2 by σ^2 yields $N \geq 384 \cdot \frac{1-\gamma}{\gamma}$ [32]. Thus, estimating probabilities having at least the order of magnitude of 10^{-5} , for instance, with a relative error of 0.1 and a confidence of 95% requires at least $N = 38,000,000$ simulation runs.

4 Numerical Reachability Analysis

Instead of indirectly approximating probabilities with statistical estimation procedures, we can use a numerical reachability analysis to solve Eq. 1. An efficient solution by applying standard numerical methods is not possible, since for realistic systems the state space of the system is extremely large. An efficient approximation is, however, possible as long as the total number of involved molecules is a manageable number. We describe a method that is based on a discretization of the process and numerically approximates the probabilities $p^{(t)}(x)$ at certain time instances.

Adaptive Uniformization. We discretize the system using *adaptive uniformization*, which has been introduced by van Moorsel [41] as a variant of *standard uniformization* [31, 34, 44, 15, 35]. Numerical methods based on uniformization have the advantage that they are numerically stable and often more efficient than other methods [37].

We inductively define a sequence S_0, S_1, \dots of subsets of the state space S of the CTMC $\{X(t)\}_{t \geq 0}$, as well as a sequence p_0, p_1, \dots of functions such that $p_k : S \rightarrow [0, 1]$ for $k = 0, 1, \dots$. Recall that y is the initial state. We define $S_0 = \{y\}$, $p_0(y) = 1$ and $p_0(x) = 0$ if $x \neq y$. For $k = 1, 2, \dots$, we inductively define S_k as follows. We choose a positive *uniformization rate* $\lambda_k \geq \max_{x \in S_k} \lambda_x$ and set

$$S_{k+1} = \{x' \in S \mid \exists x \in S_k : p_k(x) \cdot q_k(x, x') > 0\}, \quad (3)$$

where, for $x \in S$,

$$q_k(x, x') = \begin{cases} \sum_{m: u_m(x)=x'} \alpha_m(x) / \lambda_k & \text{if } x \neq x', \exists m : u_m(x) = x', \\ 0 & \text{if } x \neq x', \nexists m : u_m(x) = x', \\ 1 - \sum_{x' \in S: x' \neq x} q_k(x, x') & \text{if } x = x'. \end{cases} \quad (4)$$

For $x' \in S_{k+1}$ we set $p_{k+1}(x') = \sum_{x \in S_k} p_k(x) \cdot q_k(x, x')$ and $p_{k+1}(x) = 0$ if $x \notin S_k$.

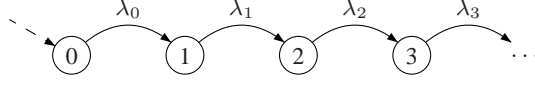


Fig. 1: The birth process of the adaptive uniformization procedure.

The value $p_k(x)$ is the probability of reaching state x after k steps in a discrete-time Markov chain $\{Y(k)\}_{k \in \mathbb{N}}$ with transition probabilities $Pr(Y(k+1) = x' \mid Y(k) = x) = q_k(x, x')$ and initial distribution $Pr(Y(0) = y) = 1$. We can reconstruct $p^{(t)}(x)$ by considering an additional process that relates steps with time. Let $\{B(t)\}_{t \geq 0}$ be a birth process with birth rates $\lambda_0, \lambda_1, \dots$, that is, B has a chain structure as illustrated in Fig. 1 and starts initially in state 0 with probability one. In [41], van Moorsel has proved that the original CTMC $\{X(t)\}_{t \geq 0}$ can be constructed from B and Y by setting $Y(B(t)) = X(t)$ if B does not explode. Since Y and B are independent, the state probability $p^{(t)}(x)$ of the original CTMC can be expressed as

$$p^{(t)}(x) = \sum_{k=0}^{\infty} Pr(Y(k) = x) \cdot Pr(B(t) = k) = \sum_{k=0}^{\infty} p_k(x) \cdot Pr(B(t) = k). \quad (5)$$

Note that in Eq. 5, there are no negative summands involved. Moreover, p_k can be computed inductively. Lower and upper summation bounds L and U can be obtained such that for each state x the truncation error

$$\begin{aligned} p^{(t)}(x) - \sum_{k=L}^U p_k(x) \cdot Pr(B(t) = k) &= \sum_{\substack{0 \leq k < L, \\ U < k < \infty}} p_k(x) \cdot Pr(B(t) = k) \leq \\ \sum_{\substack{0 \leq k < L, \\ U < k < \infty}} Pr(B(t) = k) &= 1 - \sum_{k=L}^U Pr(B(t) = k) < \epsilon \end{aligned} \quad (6)$$

can be bounded by $\epsilon > 0$. Finally, we note that from Eq. 4 it is clear that choosing the smallest possible λ_k is advantageous since this avoids high self-loop probabilities in q_k .

Standard Uniformization. Standard uniformization is a special case of adaptive uniformization where a global uniformization rate $\lambda = \lambda_0 = \lambda_1 = \dots$ has to be chosen. If each transition in the birth process occurs at a constant rate λ , the values $Pr(B(t) = k)$ follow a Poisson distribution with parameter λt . They can be calculated efficiently using the iterative procedure introduced by Fox and Glynn [8]. Standard uniformization becomes inefficient whenever λ is much larger than the exit rates $\lambda(x)$ of many states x that are involved in the computation. Note that in chemically reacting systems, the dynamics of the system may change considerably. In this case the discretization using adaptive uniformization is more efficient.

Approximate Discretization. In its standard form, adaptive uniformization is not appropriate for Markov chains that describe biochemical reaction networks for two reasons. Firstly, the size of the sets S_0, S_1, \dots grows after each step and the computational complexity for p_k becomes huge. Secondly, the birth process may explode even if the original CTMC does not. The reason is that S_k approaches S as $k \rightarrow \infty$. The latter problem can be circumvented by neglecting states that are very unlikely, that is, we

replace Eq. 3 by

$$S_{k+1} = \{x' \in S \mid \sum_{x \in S_k} p_k(x) \cdot q_k(x, x') > \Delta\}, \quad (7)$$

where Δ is a small constant. This ensures that even in the limit S_k is finite, since only a finite number of states can have a probability greater than δ . Moreover, the number of states in S_k is now manageable as long as the total number of molecules is manageable since only a comparatively small number of different values for each state variables have to be considered.

The error after k steps introduced by the threshold Δ can be calculated as $1 - \sum_{x \in S_k} p_k(x)$. Note that the error increases monotonically in k since more and more probability “gets lost”. Therefore we choose Δ several orders of magnitude smaller than the desired precision. For our experimental results in Section 5 we chose different values for Δ ranging from 10^{-15} till 10^{-8} in order to obtain different precision levels.

Approximate Solution of the Birth Process. Finally, we discuss the computation of the values $Pr(B(t) = k)$ and how truncation bounds L and R are obtained. We use standard uniformization to discretize B , since we can afford a high global uniformization rate (and thus, high self-loop probabilities) in this case. The reason is that the simple chain structure eases the discretization and the computational effort to solve the birth process is small compared to the calculation of the p_k . Similar as for Y we approximately solve B by neglecting states that are “left behind”. Informally, we use a window (a set that contains all states within a certain range) that slides from left to right to approximate $Pr(B(t) = k)$ and determines the truncation points L and R .

Approximation Error. Both, the solution of Y and B gives an underapproximation of the values $p_k(x)$ and $Pr(B(t) = k)$. Thus, summing up their product according to Eq. 5 results in an underapproximation for $p^{(t)}(x)$. The final approximation error is obtained as $\delta = 1 - \sum_{x \in S_R} p^{(t)}(x)$ where R is the right truncation bound of the birth process. The probability of states that are not in S_R is approximated with zero. Note that this includes all approximation errors, i.e., the probability that is lost during the solution of the birth process, and during all steps of the discretization because of the threshold Δ . The computational savings achieved by solving Y as well as B in the way described above are substantial. The reason is that the number of states in B and Y that are significant after k steps is several orders of magnitudes smaller than the number of all states reachable after k steps. Moreover, our experimental results show that the method yields accurate results, as the approximation error δ is small.

Iteration Over Time. First note that we can use the method described above for systems starting with arbitrary initial distributions as long as the number of states in the initial set S_0 is manageable. After computing an approximation of $p^{(t)}(x)$ for all $x \in S$ we can use it as an initial distribution for the next step to obtain an approximation for $p^{(t')}(x)$ where $t' > t$ and the step size is $t' - t$. In this way, we obtain approximations for several time instances.

Related Work. Other approaches for an approximate numerical solution of the underlying Markov chains can be found in [26, 5]. They differ from our approach in that they compute a finite projection of the state space that is based solely on the structure of

numerical approximation				Gillespie simulation		
running time	total approx. error	$ S_k $	Δ	running time	single event error	# runs
55 min 5 sec	3×10^{-6}	239792	10^{-15}	> 6000 h	10^{-8}	$> 3 \times 10^{10}$
39 min 16 sec	2×10^{-5}	187204	10^{-14}	> 500 h	10^{-7}	$> 3 \times 10^9$
25 min 2 sec	2×10^{-4}	140969	10^{-13}	67 h 22 min	10^{-6}	$> 3 \times 10^8$
15 min 41 sec	1×10^{-3}	101078	10^{-12}	6 h 44 min	10^{-5}	$> 3 \times 10^7$
6 min 33 sec	7×10^{-3}	67540	10^{-11}	40 min	10^{-4}	$> 3 \times 10^6$
3 min 12 sec	4×10^{-2}	40373	10^{-10}	4 min	10^{-3}	$> 3 \times 10^5$

Table 1: Comparison of the running times for the phage λ model.

the underlying graph. In our method, we add and neglect states in an on-the-fly fashion based on the stochastic properties of the Markov chain. Therefore, we consider a significantly smaller set of states during a certain time interval, without being less accurate. The projection algorithms include all states that are reachable within a fixed path depth. In our algorithm, for each single state, we dynamically decide if it significantly contributes to the overall solution or not. We have found this dynamic adaptation of the analysis to be essential for efficiency.

5 Experimental Results

For our experimental results, we consider two examples from biology. One is a model for the transcription regulation of a repressor protein in bacteriophage λ [13]. This protein is responsible for maintaining lysogeny of the λ virus in *E. coli* [1]. We compute the full probability distribution for different precision levels. Our second example uses the gene expression model of Ex. 1. We calculate the distribution of the time until the number of produced proteins exceeds 500.

There is no one-to-one correspondence between the statistical accuracy of the estimates that we derive via simulation and the precision of the numerical method. However, by assuming that the smallest event probability that has to be estimated is γ all results of the simulation have a “precision” of at least γ . Intuitively, we simulate often enough to reason about events that occur with a probability of at least γ . We therefore refer to γ as the *single event error* (cf. Table 1 and 2). Note that the simulation results are still subject to the statistical errors since the true values may not be covered by the confidence interval (compare Section 3.2).

The approximation error δ of the numerical method is the sum of the approximation error of *all* states in the Markov chain. Note that the probabilities of states not in S_k are underapproximated with zero and their true probabilities increase depending on how close they are to an attracting region. The error of a single state probability $p^{(t)}(x)$ is much smaller than δ but precise values for the single error are hard to obtain. A rough estimation of the single errors can be obtained by dividing the total error by the average size $|S_k|$ of the significant sets (cf. Table 1 and 2), even though δ may not be uniformly distributed on the significant set. On the other hand, δ also includes the error of insignificant states and, thus, distributes among much more states than only those in S_k .

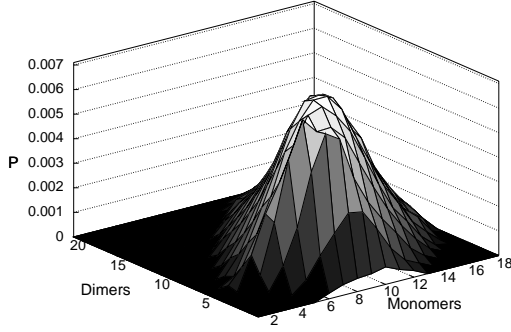


Fig. 2: Probability distribution of monomers and dimers in the phage λ model.

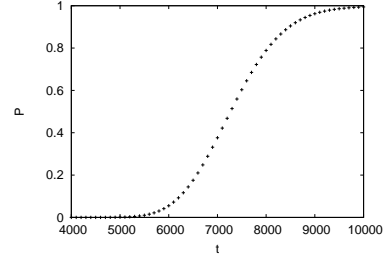


Fig. 3: Cumulative probability distribution of the time until the number of proteins reaches 500 for the first time in the gene expression example.

Phage λ Model. The Phage λ model involves 6 different species and 10 reactions. Thus, a state is a vector $x = (x_1, x_2, x_3, x_4, x_5, x_6) \in \mathbb{N}_0^6$. The transition classes $C_i = (G_i, u_i, \alpha_i)$, $1 \leq i \leq 10$ are given as follows [13].

- Production of proteins: $G_1 = \{x \in \mathbb{N}_0^6 \mid x_3 > 0\}$, $u_1(x) = (x_1 + 1, x_2, x_3, x_4, x_5, x_6)$, $\alpha_1(x) = c_1 x_3$.
- Degradation of proteins: $G_2 = \{x \in \mathbb{N}_0^6 \mid x_1 > 0\}$, $u_2(x) = (x_1 - 1, x_2, x_3, x_4, x_5, x_6)$, $\alpha_2(x) = c_2 x_1$.
- Production of mRNA: $G_3 = \{x \in \mathbb{N}_0^6 \mid x_5 > 0\}$, $u_3(x) = (x_1, x_2, x_3 + 1, x_4, x_5, x_6)$, $\alpha_3(x) = c_3 x_5$.
- Degradation of mRNA: $G_4 = \{x \in \mathbb{N}_0^6 \mid x_3 > 0\}$, $u_4(x) = (x_1, x_2, x_3 - 1, x_4, x_5, x_6)$, $\alpha_4(x) = c_4 x_3$.
- First dimer binding at operator site: $G_5 = \{x \in \mathbb{N}_0^6 \mid x_2, x_4 > 0\}$, $u_5(x) = (x_1, x_2 - 1, x_3, x_4 - 1, x_5 + 1, x_6)$, $\alpha_5(x) = c_5 x_2 x_4$. the simulation results are still subject to the statistical errors since the true values may not be covered by the confidence interval (compare Section 3.2).
- First dimer unbinding: $G_6 = \{x \in \mathbb{N}_0^6 \mid x_5 > 0\}$, $u_6(x) = (x_1, x_2 + 1, x_3, x_4 + 1, x_5 - 1, x_6)$, $\alpha_6(x) = c_6 x_5$.
- Second dimer binding at operator site: $G_7 = \{x \in \mathbb{N}_0^6 \mid x_2, x_5 > 0\}$, $u_7(x) = (x_1, x_2 - 1, x_3, x_4, x_5 - 1, x_6 + 1)$, $\alpha_7(x) = c_7 x_2 x_5$.
- Second dimer unbinding: $G_8 = \{x \in \mathbb{N}_0^6 \mid x_6 > 0\}$, $u_8(x) = (x_1, x_2 + 1, x_3, x_4, x_5 + 1, x_6 - 1)$, $\alpha_8(x) = c_8 x_6$.
- Dimerization: $G_9 = \{x \in \mathbb{N}_0^6 \mid x_1 > 1\}$, $u_9(x) = (x_1 - 2, x_2 + 1, x_3, x_4, x_5, x_6)$, $\alpha_9(x) = c_9 x_1(x_1 - 1)/2$.
- Dissociation into monomers: $G_{10} = \{x \in \mathbb{N}_0^6 \mid x_2 > 0\}$, $u_{10}(x) = (x_1 + 2, x_2 - 1, x_3, x_4, x_5, x_6)$, $\alpha_{10}(x) = c_{10} x_2$.

For c_1, \dots, c_{10} , we choose $c_1 = 0.043$, $c_2 = 0.0007$, $c_3 = 0.0715$, $c_4 = 0.0039$, $c_5 = 1.992647 \times 10^{-2}$, $c_6 = 0.4791$, $c_7 = 1.992647 \times 10^{-4}$, $c_8 = 8.765 \times 10^{-12}$, $c_9 = 8.30269 \times 10^{-2}$, and $c_{10} = 0.5$ (see [13, 5]). The initial state of the system is given by $y = (2, 6, 0, 2, 0, 0)$ and the time horizon is $t = 300$. We approximate the probability

numerical approximation				Gillespie simulation		
running time	total approx. error	$ S_k $	Δ	running time	single event error	# runs
4.2 sec	5×10^{-6}	9816	10^{-12}	> 500 h	10^{-7}	$> 3 \times 10^9$
3.6 sec	5×10^{-5}	8719	10^{-11}	> 50 h	10^{-6}	$> 3 \times 10^8$
3.0 sec	5×10^{-4}	7516	10^{-10}	5 h 3 min	10^{-5}	$> 3 \times 10^7$
2.4 sec	4×10^{-3}	6265	10^{-9}	30 min 18 sec	10^{-4}	$> 3 \times 10^6$
1.9 sec	4×10^{-2}	4939	10^{-8}	3 min sec	10^{-3}	$> 3 \times 10^5$

Table 2: Comparison of the running times for the gene expression example.

distributions of the underlying CTMC at 100 equidistant time instances. Fig. 2 shows a plot of the distribution of dimers and monomers at time instant $t = 300$. In Table 1, we list the running times of our numerical method as well as the running time of the simulation. The column with header $|S_k|$ lists the average number of states in the sets S_0, S_1, \dots and Δ is the threshold in Eq. 7.

Gene Expression. For the transition classes of the gene expression example we refer to Ex. 1. For the rate constants, we choose $c_1 = 0.05$, $c_2 = 0.0058$, $c_3 = 0.0029$, and $c_4 = 10^{-4}$, where c_3 and c_4 correspond to a half-life of 4 minutes for mRNA and 2 hours for the protein [39]. We compute the probability that at least 500 proteins are in the system at 100 equidistant time instances. Fig 3 shows the cumulative probability distribution of the time until the number of proteins reaches 500 for the first time (note that eventually the threshold of 500 is reached with probability one). In Table 2, we list the results for the gene expression example, where, as above, $|S_k|$ denotes the average number of states in the sets S_0, S_1, \dots and Δ is the threshold in Eq. 7.

Discussion. Even if we consider the total approximation error δ as a rough bound for the single error of each state probability, thus favoring simulation, the speed-up factor of the numerical approximation is large, especially if the precision increases. The necessary precision level up to which probability distributions are approximated may depend on the system under study. It is, however, important to note that the occurrence of rare biochemical events can have important effects. For instance, the spontaneous, epigenetic switching rate from the lysogenic state to the lytic state in phage λ -infected *E. coli* is experimentally estimated to be in the order of 10^{-7} per cell per generation [21].

6 Conclusion

We have demonstrated that, for the computation of event probabilities, a numerical reachability analysis provides an efficient alternative to simulation-based methods.

Even though simulation is widely used, the advantages of numerical methods increase as more sophisticated techniques become available. They reduce the computational effort, especially if accurate results are desired. Moreover, for the calibration of parameters many instances of the model have to be solved and in this case short running times for a single solution are necessary.

Until now we have analyzed examples of intrinsically stochastic systems that have been published in the literature. As future work, we are planning to apply our numerical

reachability algorithm in collaboration with experimentalists working on new stochastic models. Moreover, we are planning to combine our numerical method with parameter estimation techniques.

Standard numerical reachability analysis methods are inefficient for large state spaces (in the case of high dimension and/or many molecules) and inapplicable for unbounded state spaces, and thus one resorts to simulation. We have demonstrated that certain optimization techniques from computer science - localization, on the fly abstraction - put many examples within the reach of numerical reachability analysis. Indeed, when high accuracy is required these methods outperform simulation-based techniques.

References

1. A. Arkin, J. Ross, and H. H. McAdams. Stochastic kinetic analysis of developmental pathway bifurcation in phage λ -infected E. coli cells. *Genetics*, 149:1633–1648, 1998.
2. N. Barkai and S. Leibler. Biological rhythms: Circadian clocks limited by noise. *Nature*, 403:267–268, 2000.
3. W. J. Blake, M. Kaern, C. R. Cantor, and J. J. Collins. Noise in eukaryotic gene expression. *Nature*, 422:633–637, 2003.
4. P. Bremaud. *Markov Chains*. Springer, 1998.
5. K. Burrage, M. Hegland, F. Macnamara, and R. Sidje. A Krylov-based finite state projection algorithm for solving the chemical master equation arising in the discrete modelling of biological systems. In *Proc. of the Markov 150th Anniversary Conference*, pages 21–38. Boson Books, 2006.
6. M. B. Elowitz, M. J. Levine, E. D. Siggia, and P. S. Swain. Stochastic gene expression in a single cell. *Science*, 297:1183–1186, 2002.
7. N. Fedoroff and W. Fontana. Small numbers of big molecules. *Science*, 297:1129–1131, 2002.
8. B. L. Fox and P. W. Glynn. Computing Poisson probabilities. *Communications of the ACM*, 31(4):440–445, 1988.
9. D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81(25):2340–2361, 1977.
10. D. T. Gillespie. *Markov Processes*. Academic Press, N. Y., 1992.
11. D. Gonze, J. Halloy, and A. Goldbeter. Robustness of circadian rhythms with respect to molecular noise. *PNAS, USA*, 99(2):673–678, 2002.
12. D. Gonze, J. Halloy, and A. Goldbeter. Stochastic models for circadian oscillations: Emergence of a biological rhythm. *Quantum Chemistry*, 98:228–238, 2004.
13. J. Goutsias. Quasiequilibrium approximation of fast reaction kinetics in stochastic biochemical systems. *J. Chem. Phys.*, 122(18):184102, 2005.
14. J. Hasty, J. Pradines, M. Dolnik, and J. J. Collins. Noise-based switches and amplifiers for gene expression. *PNAS USA*, 97:2075, 2000.
15. A. Hellander. Efficient computation of transient solutions of the chemical master equation based on uniformization and quasi-Monte carlo. *J. Chem. Phys.*, 128(15):154109, 2008.
16. D. A. Henderson, R. J. Boys, C. J. Proctor, and D. J. Wilkinson. Linking systems biology models to data: a stochastic kinetic model of p53 oscillations. In A. O’Hagan and M. West, editors, *Handbook of Applied Bayesian Analysis*. Oxford University Press, 2009.
17. T. Henzinger, M. Mateescu, and V. Wolf. Sliding window abstraction for infinite Markov chains. In *Proc. CAV, LNCS*. Springer, 2009. To appear.
18. N. G. van Kampen. *Stochastic Processes in Physics and Chemistry*. Elsevier, 3rd edition, 2007.

19. A. Kierzek, J. Zaim, and P. Zielenkiewicz. The effect of transcription and translation initiation frequencies on the stochastic fluctuations in prokaryotic gene expression. *Journal of Biological Chemistry*, 276(11):8165–8172, 2001.
20. A. Law and D. Kelton. *Simulation Modelling and Analysis*. McGraw-Hill Education, 2000.
21. J. W. Little, D. P. Shepley, and D. W. Wert. Robustness of a gene regulatory circuit. *The EMBO Journal*, 18(15):4299–4307, 1999.
22. R. Losick and C. Desplan. Stochasticity and Cell Fate. *Science*, 320(5872):65–68, 2008.
23. H. Maamar, A. Raj, and D. Dubnau. Noise in gene expression determines cell fate in *Bacillus subtilis*. *Science*, 317(5837):526 – 529, 2007.
24. H. H. McAdams and A. Arkin. Stochastic mechanisms in gene expression. *PNAS, USA*, 94:814–819, 1997.
25. H. H. McAdams and A. Arkin. It’s a noisy business! *Trends in Genetics*, 15(2):65–69, 1999.
26. B. Munsky and M. Khammash. The finite state projection algorithm for the solution of the chemical master equation. *J. Chem. Phys.*, 124:044144, 2006.
27. E. M. Ozbudak, M. Thattai, I. Kurtser, A. D. Grossman, and A. van Oudenaarden. Regulation of noise in the expression of a single gene. *Nature Genetics*, 31(1):69 – 73, 2002.
28. P. Patel, B. Arcangioli, S. Baker, A. Bensimon, and N. Rhind. DNA replication origins fire stochastically in fission yeast. *Mol Biol Cell*, 17:308–316, 2006.
29. J. Paulsson. Summing up the noise in gene networks. *Nature*, 427(6973):415–418, 2004.
30. C. Rao, D. Wolf, and A. Arkin. Control, exploitation and tolerance of intracellular noise. *Nature*, 420(6912):231–237, 2002.
31. W. Sandmann. Stochastic simulation of biochemical systems via discrete-time conversion. In *Proceedings of the 2nd Conference on Foundations of Systems Biology in Engineering*, pages 267–272. Fraunhofer IRB Verlag, 2007.
32. W. Sandmann and C. Maier. On the statistical accuracy of stochastic simulation algorithms implemented in Dizzy. In *Proc. WCSB*, pages 153–156, 2008.
33. W. Sandmann and V. Wolf. A computational stochastic modeling formalism for biological networks. In *Enformatika Transactions on Engineering, Computing and Technology*, volume 14, pages 132–137, 2006.
34. W. Sandmann and V. Wolf. Computational probability for systems biology. In *Proc. FMSB*, volume 5054 of *LNCS*, pages 33–47. Springer, 2008.
35. R. Sidje, K. Burrage, and S. MacNamara. Inexact uniformization method for computing transient distributions of Markov chains. *SIAM J. Sci. Comput.*, 29(6):2562–2580, 2007.
36. R. Srivastava, L. You, J. Summers, and J. Yin. Stochastic vs. deterministic modeling of intracellular viral kinetics. *Journal of Theoretical Biology*, 218:309–321, 2002.
37. W. J. Stewart. *Introduction to the Numerical Solution of Markov Chains*. Princeton University Press, 1995.
38. P. S. Swain, M. B. Elowitz, and E. D. Siggia. Intrinsic and extrinsic contributions to stochasticity in gene expression. *PNAS, USA*, 99(20):12795–12800, 2002.
39. M. Thattai and A. van Oudenaarden. Intrinsic noise in gene regulatory networks. *PNAS, USA*, 98(15):8614–8619, July 2001.
40. T. E. Turner, S. Schnell, and K. Burrage. Stochastic approaches for modelling in vivo reactions. *Computational Biology and Chemistry*, 28:165–178, 2004.
41. A. van Moorsel and W. Sanders. Adaptive uniformization. *ORSA Communications in Statistics: Stochastic Models*, 10(3):619–648, 1994.
42. A. Warmflash and A. Dinner. Signatures of combinatorial regulation in intrinsic biological noise. *PNAS*, 105(45):17262–17267, 2008.
43. D. J. Wilkinson. *Stochastic Modelling for Systems Biology*. Chapman & Hall, 2006.
44. J. Zhang, L. T. Watson, and Y. Cao. A modified uniformization method for the solution of the chemical master equation., 2007. TR-07-31, Computer Science, Virginia Tech.